COMPUTER SCIENCE– NEW (083)

PRACTICAL FILE

2020-2021

NAME: Ujjwal Sharma

CLASS: 12

SECTION: S1

ROLL NO: 40

INDEX

17)	WAP to create a matrix with m number of rows and n number of columns. Display the elements in a matrix format. Note: The number of rows and columns will be decided at runtime of the program.		**Done**
18)	WAP to enter any number and display it in words. Ex 123 - One Two Three		**Done**
19)	Create a dictionary with the roll number, name and marks of n students in a class and display the names of students who have marks above 75.		**Done**
20)	WAP to take the number of students as input, then ask marks for five subjects as English, Physics, Chemistry, Maths and Computer. If the total marks for any student are less than 200, then print he failed or else print passed. Use the dictionary to store the student name as key and marks as value.		**Done**
21)	WAP to input employee number and name for 'N' employees and display all information in ascending order of their employee number.		**Done**
22)	WAP to create a dictionary called Census with country name and population. Using two functions, perform the following: Search_Country() – Enter the country name and displays its respective population. Delete_Country() – Enter the country name and delete its respective population.		**Done**
23)	Given a dual Tuple list, the task is to write a python program to convert second element to negative magnitude of each tuple and first element to positive magnitude of each tuple. **Input :** test_list = [(3, -1), (-4, -3), (1, 3), (-2, 5), (-4, 2), (-9, -3)] **Output :** [(3, -1), (4, -3), (1, -3), (2, -5), (4, -2), (9, -3)]		**Done**
24)	Write a Python program to check whether an element exists within a tuple.		**Done**
25)	Write a Python program to compute the sum of all the elements of each tuple stored inside a list of tuples. Original list of tuples: [(1, 2), (2, 3), (3, 4)] Sum of all the elements of each tuple stored inside the said list of tuples: [3, 5, 7]		**Done**
26)	Read a text file line by line and display each word separated by a #. Read a text file and display the number of vowels/ consonants/ uppercase/ lowercase characters in the file.		**Done**
27)	Create a binary file with name and roll number. Search for a given roll number and display the name, if not found display appropriate message.		**Done**

28)	Create a binary file with roll number, name and marks. Input a roll number and update the marks.		**Done**
29)	Remove all the lines that contain the character `a' in a file and write it to another file.		**Done**
30)	Write a python program to search and display the record of the student from a binary file "Student.dat" containing students records (Rollno, Name and Marks). The user will enter roll number of the student to be searched.		**Done**
31)	WAP using function merge(file1,file2) to merge two files and obtain a third file called "NEW.DAT". The file1 and file2 are two files, which are pass as arguments. Also, display the content of third file.		**Done**
32)	Write a menu driven program where you are getting product related details in a file product.txt. menu options are 1. Add Details 2. Search Details 3. Show Details 4. Exit		**Done**
33)	WAP to delete the file(s), which you want, should no longer exist in your computer. The file name should be entered at the time of program execution. If the entered file name not exist, display a proper message on screen.		**Done**
34)	A file emp.dat contains data attributes like : ecode, name and salary. Give function definitions to do the following a) Write the data of an employee. b) Read the employee data and display all the objects on the screen where salary is between 20000 and 30000.		**Done**
35)	Write a menu driven program to add and manipulate data from customer.csv file. Give function to do the following: 1. Add Customer Details 2. Search Customer Details 3. Remove Customer Details 4. Display all the Customer Details 5. Exit		**Done**
36)	Write a program to create a queue called Doctor to perform the basic operations on queue using list. The list contains two data fields: Docid and Docname. Write the following functions: InsertDoc() – To push the data values into the list Docinfo DeleteDoc() – To remove the data value from the list Docinfo ShowDoc(): - To display data value for all Docinfo.		**Done**

37)	Write two functions queins() to insert and quedel() to delete elements for customeer information, i.e. custno, cname using list.		**Done**
38)	Write a program to create a stack called Product to perform the basic operations on stack using list. The list contains two data fields: ProductId and ProductName. Write the following functions: InsertProd() – To push the data values into the list Docinfo DeleteProd() – To remove the data value from the list Docinfo ShowProd(): - To display data value for all Docinfo.		**Done**
39)	Write a menu-based program to perform the operation on queue in python.		**Done**
40)	Creating and manipulating database and table structure in SQL – Create, Alter, Drop, Show and Describe. (a) Display all the available databases available in SQL. (b) Create database office and open it. (c) Display all the available tables of Office database. (d) Create a table Emp with following fields. Use appropriate constraints. Empno, Empname, Desig, Hiredate, Salary, Deptno (e) Add Mobile and Email field in table. (f) Remove Mobile field from the table. (g) Display the structure of Emp table. (h) Remove the Emp table structure and recreate it.		**Done**
41)	Inserting, projecting and manipulating records – Insert, Select, Update, Delete (a) Add 10 records in Emp table. (b) Display all the records of Emp table. (c) Increase all the salary of emp by 500. (d) Delete all the employee details working in department no 50.		**Done**
42)	**Queries using DISTINCT, BETWEEN, IN, LIKE, IS NULL, ORDER BY, GROUP BY, HAVING** (a) Display the number of departments. Each department should be displayed once. (b) Find the name and salary of those employees whose salary is between 35000 and 40000. (c) Find the name of those employees who live in guwahati, surat or jaipur city. (d) Display the name of those employees whose name starts with 'M'.		**Done**

	(e) List the name of employees not assigned to any department. (f) Display the list of employees in descending order of employee code. (g) Find the average salary at each department. (h) Find maximum salary of each department and display the name of that department which has maximum salary more than 39000.		
43)	**Queries for Aggregate functions- SUM( ), AVG( ), MIN( ), MAX( ), COUNT( )** a. Find the average salary of the employees in employee table. b. Find the minimum salary of a female employee in EMPLOYEE table. c. Find the maximum salary of a male employee in EMPLOYEE table. d. Find the total salary of those employees who work in Guwahati city.		**Done**
44)	Queries for Joining Tables – Cartesian join & Equi join		**Done**
45)	Write a program to connect Python with MySQL using database connectivity and perform the following operations on data in database: Fetch, Update and delete the data.		**Done**

CODE AND ANSWERS:

1.

```python
# Determine whether a number is a perfect number, an Armstrong number or a palindrome.

def perfect_number(n):
    sum1 = 0
    for i in range(1, n):
        if(n % i == 0):
            sum1 = sum1 + i
    if (sum1 == n):
        return True
    else:
        return False


def armstrong(n):
    sum = 0
    temp = n
    while temp > 0:
        digit = temp % 10
        sum += digit ** 3
        temp //= 10
    if n == sum:
        return True
    else:
        return False


def palindrome(n):
    temp = n
    rev = 0
    while(n > 0):
        dig = n % 10
        rev = rev*10+dig
        n = n//10
    if(temp == rev):
        return True
    else:
        return False


k = True
while k == True:
    n = int(input("Enter a number: "))
    if perfect_number(n) == True:
        print(n, " is a Perfect number!")
    elif armstrong(n) == True:
        print(n, " is an Armstrong number!")
    elif palindrome(n) == True:      You, seconds ago • Uncommitted changes
        print("The number is a palindrome!")
    else:
        print("The number is a Unknow!")
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a number: 153
153  is an Armstrong number!
Do you want to try again.(y/n): y
Enter a number: 12321
The number is a palindrome!
Do you want to try again.(y/n): y
Enter a number: 2
The number is a palindrome!
Do you want to try again.(y/n): n
```

2.

```python
# Input a number and check if the number is a prime or composite number.

k = True
while k == True:
    num = int(input('Enter a number: '))
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                print(num, "is NOT a prime number")
                break
        else:
            print(num, "is a PRIME number")
    elif num == 0 or 1:
        print(num, "is a neither prime NOR composite number")
    else:
        print(num, "is NOT a prime number it is a COMPOSITE number")
    option = input('Do you Want to try again(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a number: 13
13 is a PRIME number
Do you Want to try again(y/n): y
Enter a number: 12
12 is NOT a prime number
Do you Want to try again(y/n): y
Enter a number: 1
1 is a neither prime NOR composite number
Do you Want to try again(y/n): y
Enter a number: 10
10 is NOT a prime number
Do you Want to try again(y/n): n
```

3.

```
# Display the terms of a Fibonacci series.

k = True
while k == True:
    nterms = int(input("How many terms: "))
    n1, n2 = 0, 1
    count = 0
    if nterms <= 0:
        print("Please enter a positive integer")
        continue
    elif nterms == 1:
        print("Fibonacci sequence upto", nterms, ":")
        print(n1)
    else:
        print("Fibonacci sequence:")
        while count < nterms:
            print(n1)
            nth = n1 + n2
            n1 = n2
            n2 = nth
            count += 1
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
How many terms: 4
Fibonacci sequence:
0
1
1
2
Do you want to try again.(y/n): n
```

4.

```python
# Compute the greatest common divisor and least
# common multiple of two integers.


def gcd(x, y):
    while y > 0:
        x, y = y, x % y
    return x


def lcm(x, y):
    lcm = (x*y)//gcd(x, y)
    return lcm


k = True
while k == True:
    x = int(input('Enter a number: '))
    y = int(input('Enter a number: '))
    print(f'The GCD is {gcd(x, y)}')
    print(f'The LCM is {lcm(x, y)}')
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a number: 13
Enter a number: 26
The GCD is 13
The LCM is 26
Do you want to try again.(y/n): n
```

5.

```
# Count and display the number of vowels,
# consonants, uppercase, lowercase characters in string


def countCharacterType(s):
    vowels = 0
    consonant = 0
    lowercase = 0
    uppercase = 0

    for i in range(0, len(s)):
        ch = s[i]
        if ((ch ≥ 'a' and ch ≤ 'z') or
                (ch ≥ 'A' and ch ≤ 'Z')):
            if ch.islower():
                lowercase += 1
            if ch.isupper():
                uppercase += 1
            ch = ch.lower()

            if (ch == 'a' or ch == 'e' or ch == 'i'
                    or ch == 'o' or ch == 'u'):
                vowels += 1
            else:
                consonant += 1

    print("Vowels:", vowels)
    print("Consonant:", consonant)
    print("LowerCase:", lowercase)
    print("UpperCase:", uppercase)


k = True
while k == True:
    s = input('Enter a string: ')
    countCharacterType(s)
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a string: hi! My name is Ujjwal Sharma.
Vowels: 8
Consonant: 14
LowerCase: 19
UpperCase: 3
Do you want to try again.(y/n): n
```

6.

```
# Input a string and determine whether it is
# a palindrome or not; convert the case of characters in a string.


def is_palindrome(s):
    return s == s[::-1]


k = True
while k == True:
    s = input("Enter a string: ")
    if is_palindrome(s):
        print('Entered Srting is Palindrom string!')
        print(f'Converted case string is: {s.swapcase()}')
    else:
        print('Entered Srting is not Palindrom string!')
        print(f'Converted case string is: {s.swapcase()}')

    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a string: Hi! My name is Ujjwal Sharma.
Entered Srting is not Palindrom string!
Converted case string is: hI! mY NAME IS uJJWAL sHARMA.
Do you want to try again.(y/n): y
Enter a string: lol
Entered Srting is Palindrom string!
Converted case string is: LOL
Do you want to try again.(y/n): n
```

7.

```python
# write a program for binary search in python

def binary_search(arr, x):
    low = 0
    high = len(arr) - 1
    mid = 0
    while low <= high:
        mid = (high + low) // 2
        if arr[mid] < x:
            low = mid + 1
        elif arr[mid] > x:
            high = mid - 1
        else:
            return mid
    return -1


k = True
while k == True:
    arr = list(
        map(int, input('Enter a list of no.s(separated by commas): ').split(',')))
    x = int(input('Enter a no. to search: '))
    print(arr)
    result = binary_search(arr, x)
    if result == -1:
        print("Element is not present in array")
    else:
        print(f'Element is present at index {result}')
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a list of no.s(separated by commas): 54,2,6,1,8,45,34,09
Enter a no. to search: 45
[54, 2, 6, 1, 8, 45, 34, 9]
Element is present at index 5
Do you want to try again.(y/n): n
```

8.

```
# Write a program to generate random numbers between
# 1 to 6 and check whether a user won a lottery or not.


def random_game():
    import random
    num = random.randint(1, 6)
    a = int(input("Enter a Random numbers between(1/6): "))
    if a == num:
        print("You won the Game!")
    else:
        print('Better luck next time..')


k = True
while k == True:
    random_game()
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a Random numbers between(1/6): 4
Better luck next time..
Do you want to try again.(y/n): y
Enter a Random numbers between(1/6): 3
Better luck next time..
Do you want to try again.(y/n): y
Enter a Random numbers between(1/6): 3
You won the Game!
Do you want to try again.(y/n): n
```

9.

```
# Write a program to create a library      You, 6 days ago • all
# in python and import it in a program.

import example

k = True
while k == True:
    example.hello()
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False

def hello():      You, 6 c
    print('Hello World!')
```

```
Hello World!
Do you want to try again.(y/n): n
```

10.

```
# write a program for linear search in python


def linearsearch(arr, x):
    for i in range(len(arr)):
        if arr[i] == x:
            return i
    return -1


k = True
while k == True:
    arr = list(
        map(int, input('Enter a list of numbers(separated by commas): ').split(',')))
    x = int(input('Enter a number: '))
    print("element found at index "+str(linearsearch(arr, x)))
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a list of numbers(separated by commas): 4,2,8,23,56,0,1,34
Enter a number: 0
element found at index 5
Do you want to try again.(y/n): n
```

11.

```python
# Write a program for bubble sort


def bubbleSort(arr):
    n = len(arr)
    for i in range(n-1):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]


k = True
while k == True:
    arr = list(
        map(int, input('Enter a list of no.s(separated by commas): ').split(',')))
    bubbleSort(arr)
    print(arr)
    option = input('Do you want to try again(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a list of no.s(separated by commas): 76,32,65,2,675,23
[2, 23, 32, 65, 76, 675]

Do you want to try again(y/n): n
```

12.

```
# Input a list / tuple of elements, search
# for a given element in the list/tuple.


k = True
while k == True:
    arr = list(
        map(int, input('Enter a list of no.s(separated by commas): ').split(',')))
    x = int(input('Enter a no. to search: '))
    for i in range(0, len(arr) - 1):
        if arr[i] == x:
            print(f'Entered no. is in the list, At index number: {arr[i]}')
    option = input('Do you want to try again(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a list of no.s(separated by commas): 12,65,2,7,2,76,45
Enter a no. to search: 7
Entered no. is in the list, At index number: 7
Do you want to try again(y/n): n
```

13.

```
# Input a list of numbers and test if a number is equal to the sum of
# the cubes of its digits. Find the smallest and largest such number
# from the given list of numbers.


def Armstrong(arr):
    for num in arr:
        sum = 0
        temp = num
        while temp > 0:
            digit = temp % 10
            sum += digit ** 3
            temp //= 10
        if num == sum:
            print(f'Armstrong no. is: {num}')
    arr.sort()      You, seconds ago • Uncommitted changes
    smallest = arr[0]
    largest = arr[-1]
    print(f'smallest value is {smallest}')
    print(f'largest value is {largest}')


k = True
while k == True:
    arr = list(
        map(int, input('Enter a list of no.s(separated by commas): ').split(',')))
    Armstrong(arr)
    option = input('Do you want to try again.(y/n): ').lower()
    if option == 'y':
        continue
    else:
        k = False
```

```
Enter a list of no.s(separated by commas): 153,0,2,567
Armstrong no. is: 153
Armstrong no. is: 0
smallest value is 0
largest value is 567
Do you want to try again.(y/n): n
```

14.

```python
# Write a program to input five students name and their total marks
# in first semester. Find the highest mark and the name of the student.

student_list = {}
for _ in range(0, 5):
    name = input("Enter Your Name: ")
    marks = int(input('Enter your marks: '))
    student_list[marks] = name

highest = sorted(student_list.items())
print(f'{highest[-1][1]} has the highest mark: {highest[-1][0]}')
```

```
Enter Your Name: ujjwal
Enter your marks: 490
Enter Your Name: bala
Enter your marks: 500
Enter Your Name: mrk
Enter your marks: 400
Enter Your Name: sid
Enter your marks: 480
Enter Your Name: garvit
Enter your marks: 200
bala has the highest mark: 500
```

15.

```
# Write a program to create a list by entering countries and respective
# capital and population. The program should accept the name of a country
# as an input and print the corresponding capital name and population as
# output. Otherwise, the program should print an appropriate message if
# the country is not found in the list. Also, display the details of the
# list in descending order.

country = {}


def add_country():
    n = int(input("Enter number of Country's: "))
    for i in range(n):
        name = input("Country Name: ")
        capital = input("Country Capital: ")
        population = int(input("Country Population: "))
        country[name] = [capital, population]


def Search_Country():
    name = input("Enter Country name: ")
    for keys, values in country.items():
        if keys == name:
            print(f'Capital: {values[0]}')
            print(f'Population: {values[1]}')


k = True
while k == True:
    print('''
1.Add Country Details
2.Search Country Details
3.See Country Details
4.Exit
    ''')
    option = int(input('Enter your option(1/4): '))
    if 0 < option <= 3:
        if option == 1:
            add_country()
        elif option == 2:
            Search_Country()
        elif option == 3:
            print(country)
    elif option == 4:
        k = False
    else:
        print("Invalid Option!")
```

```
1.Add Country Details
2.Search Country Details
3.See Country Details
4.Exit

Enter your option(1/4): 1
Enter number of Country's: 2
Country Name: india
Country Capital: delhi
Country Population: 1000000
Country Name: europe
Country Capital: london
Country Population: 1000

1.Add Country Details
2.Search Country Details
3.See Country Details
4.Exit

Enter your option(1/4): 2
Enter Country name: india
Capital: delhi
Population: 1000000

1.Add Country Details
2.Search Country Details
3.See Country Details
4.Exit

Enter your option(1/4): 3
{'india': ['delhi', 1000000], 'europe': ['london', 1000]}

1.Add Country Details
2.Search Country Details
3.See Country Details
4.Exit

Enter your option(1/4): 4
```

16.

```
# Write a program using function called insertionSort(Num) to arrange a
# list of integer elements in ascending order using insertion sort technique.

def insertionSort(num):
    for i in range(1, len(arr)):
        key = arr[i]
        j = i-1
        while j ≥ 0 and key < arr[j]:
            arr[j + 1] = arr[j]
            j -= 1
        arr[j + 1] = key


arr = list(
    map(int, input('Enter a list of no.s(separated by commas): ').split(',')))
insertionSort(arr)
print(arr)
```

```
Enter a list of no.s(separated by commas): 12,5,23,7,34,4,2,1
[1, 2, 4, 5, 7, 12, 23, 34]
```

17.

```
# WAP to create a matrix with m number of rows and n number
# of columns. Display the elements in a matrix format.
# Note: The number of rows and columns will be decided at
# runtime of the program.

row_num = int(input("Input number of rows: "))
col_num = int(input("Input number of columns: "))
multi_list = [[0 for col in range(col_num)] for row in range(row_num)]

for row in range(row_num):
    for col in range(col_num):
        multi_list[row][col] = row*col

print(multi_list)
```

```
Input number of rows: 3
Input number of columns: 4
[[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]
```

18.

```
# WAP to enter any number and display it in words.
# Ex 123 - One Two Three

num = {'1': 'One', '2': 'Two', '3': 'Three', '4': 'Four', '5': 'Five',
       '6': 'Six', '7': 'Seven', '8': 'Eight', '9': 'Nine', '0': 'Zero'}

n = input('Enter a number: ')
num_in_word = ''
for i in n:
    for keys, values in num.items():
        if i == keys:
            num_in_word += f'{values} '

print(f'This {n} in Words: {num_in_word}')
```

```
Enter a number: 1324
This 1324 in Words: One Three Two Four
```

19.

```
# Create a dictionary with the roll number, name and marks
# of n students in a class and display the names of students
# who have marks above 75.


n = int(input("Enter number of students: "))
result = {}
for i in range(n):
    print("Enter Details of student No.", i+1)
    rno = int(input("Roll No: "))
    name = input("Name: ")
    marks = int(input("Marks: "))
    result[rno] = [name, marks]       You, 2 days ago • added c
for student in result:
    if result[student][1] > 75:
        print(f'{result[student][0]} has marks more than 75')
```

```
Enter number of students: 3
Enter Details of student No. 1
Roll No: 1
Name: ujjwak
Marks: 90
Enter Details of student No. 2
Roll No: 2
Name: garvit
Marks: 70
Enter Details of student No. 3
Roll No: 3
Name: mrk
Marks: 76
ujjwak has marks more than 75
mrk has marks more than 75
```

20.

```
# Physics, Chemistry, Maths and Computer. If
# the total marks for any student are less than
# 200, then print he failed or else print passed.
# Use the dictionary to store the student name
# as key and marks as value.


n = int(input("Enter number of students: "))
result = {}
for i in range(n):
    name = input("Enter student name: ")
    english = int(input("Please enter English Marks: "))
    math = int(input("Please enter Math score: "))
    computers = int(input("Please enter Computer Marks: "))
    physics = int(input("Please enter Physics Marks: "))
    chemistry = int(input("Please enter Chemistry Marks: "))
    total = english + math + computers + physics + chemistry
    result[name] = total

for keys, values in result.items():
    if values ⩾ 200:
        print(f'{keys}: Passed')
    else:
        print(f'{keys}: Failed')
```

```
Enter number of students: 2
Enter student name: ujjwal
Please enter English Marks: 90
Please enter Math score: 90
Please enter Computer Marks: 90
Please enter Physics Marks: 90
Please enter Chemistry Marks: 90
Enter student name: mrk
Please enter English Marks: 30
Please enter Math score: 30
Please enter Computer Marks: 30
Please enter Physics Marks: 30
Please enter Chemistry Marks: 30
ujjwal: Passed
mrk: Failed
```

21.

```
# WAP to input employee number and name for 'N'
# employees and display all information in ascending
# order of their employee number

n = int(input("Enter number of Employee: "))
emp = {}
for i in range(n):
    print("Enter Details of Employee No.", i+1)
    Empno = int(input("Employee No: "))
    name = input("Name: ")
    emp[Empno] = name
ascending = list((emp.items()))
ascending.sort()
print(ascending)
```

```
Enter number of Employee: 3
Enter Details of Employee No. 1
Employee No: 2
Name: ujjwal
Enter Details of Employee No. 2
Employee No: 1
Name: mrk
Enter Details of Employee No. 3
Employee No: 7
Name: garvit
[(1, 'mrk'), (2, 'ujjwal'), (7, 'garvit')]
```

22.

```
# WAP to create a dictionary called Census with
# country name and population. Using two functions,
# perform the following:
# Search_Country() – Enter the country name and
# displays its respective population.
# Delete_Country() – Enter the country name and
# delete its respective population.


def Search_Country():
    name = input("Enter Country name: ")
    for keys, values in country.items():
        if keys == name:
            print(f'{keys} has {values} Population.')


def Delete_Country():
    name = input("Enter Country Name: ")
    del country[name]
    print('Country deleted')


k = True
while k == True:
    print('''
1.Add Country Details
2.Search Country Details
3.Delete Country Details
4.See Country Details
5.Exit
    ''')
    option = int(input('Enter your option(1/5): '))
    if 0 < option <= 4:
        if option == 1:
            n = int(input("Enter number of Country: "))
            country = {}
            for i in range(n):
                print("Enter Details of Country No.", i+1)
                name = input("Country Name: ")
                population = int(input("Country Population: "))
                country[name] = population
        elif option == 2:
            Search_Country()
        elif option == 3:
            Delete_Country()
        elif option == 4:
            print(country)
    elif option == 5:
        k = False
    else:
        print("Invalid Option!")
```

```
1.Add Country Details
2.Search Country Details
3.Delete Country Details
4.See Country Details
5.Exit

Enter your option(1/5): 1
Enter number of Country: 2
Enter Details of Country No. 1
Country Name: India
Country Population: 9999999
Enter Details of Country No. 2
Country Name: USA
Country Population: 10

1.Add Country Details
2.Search Country Details
3.Delete Country Details
4.See Country Details
5.Exit

Enter your option(1/5): 2
Enter Country name: USA
USA has 10 Population.

1.Add Country Details
2.Search Country Details
3.Delete Country Details
4.See Country Details
5.Exit

Enter your option(1/5): 3
Enter Country Name: USA
Country deleted

1.Add Country Details
2.Search Country Details
3.Delete Country Details
4.See Country Details
5.Exit

Enter your option(1/5): 4
{'India': 9999999}

1.Add Country Details
2.Search Country Details
3.Delete Country Details
4.See Country Details
5.Exit

Enter your option(1/5): 5
```

23.

```
# Given a dual Tuple list, the task is to write
# a python program to convert second element to
# negative magnitude of each tuple and first element
# to positive magnitude of each tuple.


test_list = [(3, -1), (-4, -3), (1, 3), (-2, 5), (-4, 2), (-9, -3)]
print("The original list is : " + str(test_list))
res = []
for sub in test_list:
    res.append((abs(sub[0]), -abs(sub[1])))
print("Updated Tuple list : " + str(res))
```

```
The original list is : [(3, -1), (-4, -3), (1, 3), (-2, 5), (-4, 2), (-9, -3)]
Updated Tuple list : [(3, -1), (4, -3), (1, -3), (2, -5), (4, -2), (9, -3)]
```

24.

```
# Write a Python program to check whether an element
# exists within a tuple.

arr = tuple(
    map(str, input('Enter a list of numbers(separate by comma): ').split(',')))
element = input('Enter element to check: ')
print(element in arr)
```

```
Enter a list of numbers(separate by comma): 2,1,4,3,6,9,8,7
Enter element to check: 6
True
```

25.

```
# Write a Python program to compute the sum of
# all the elements of each tuple stored inside
# a list of tuples.

# Original list of tuples:
# [(1, 2), (2, 3), (3, 4)]

# Sum of all the elements of each tuple stored
# inside the said list of tuples:
# [3, 5, 7]


def test(lst):
    result = map(sum, lst)
    print(f'Sum is: {list(result)}')


lst = [(1, 2), (2, 3), (3, 4)]
test(lst)
```

```
Sum is: [3, 5, 7]
```

26.

```
# Read a text file line by line and display each
# word separated by a #. Read a text file and display
# the number of vowels/ consonants/ uppercase/ lowercase
# characters in the file.


def word_separator():
    with open("26a.txt", "r") as f:
        doc = f.readlines()
        for i in doc:
            words = i.split()
            for a in words:
                print(a + "#", end=' ')
            print()


word_separator()


def countCharacterType():
    vowels = 0
    consonant = 0
    lowercase = 0
    uppercase = 0
    with open('26a.txt', "r") as f:
        s = f.read().split()
        for i in range(0, len(s)):
            ch = s[i]
            if ((ch ≥ 'a' and ch ≤ 'z') or (ch ≥ 'A' and ch ≤ 'Z')):
                if ch.islower():
                    lowercase += 1
                else:
                    uppercase += 1
                ch = ch.lower()

                if (ch == 'a' or ch == 'e' or ch == 'i'
                        or ch == 'o' or ch == 'u'):
                    vowels += 1
                else:
                    consonant += 1

        print("Vowels:", vowels)
        print("Consonant:", consonant)      You, 2 days ago • added code
        print("LowerCase:", lowercase)
        print("UpperCase:", uppercase)


countCharacterType()
```

TEXT FILE (26A.TXT)

```
Lorem Ipsum is simply dummy text of the printing and typesetting industry.
Lorem Ipsum has been the industry's standard dummy text ever since the 1500s,
when an unknown printer took a galley of type and scrambled it to make a
type specimen book.
```

```
Lorem# Ipsum# is# simply# dummy# text# of# the# printing# and# typesetting# industry.#
Lorem# Ipsum# has# been# the# industry's# standard# dummy# text# ever# since# the# 1500s,#
when# an# unknown# printer# took# a# galley# of# type# and# scrambled# it# to# make# a#
type# specimen# book.#
Vowels: 2
Consonant: 40
LowerCase: 38
UpperCase: 4
```

27.

```python
# Create a binary file with name and roll number.
# Search for a given roll number and display the name,
# if not found display appropriate message.

import pickle


def Add_data():
    data = []
    with open("tele.dat", 'ab') as f:
        a = int(input('Enter how many record to add: '))
        for _ in range(0, a):
            roll = int(input("Enter Student Roll no.: "))
            name = input("Enter Student Name: ")
            temp = (roll, name)
            data.append(temp)
        pickle.dump(data, f)


def Show_data():
    k = True
    with open("tele.dat", 'rb') as f:
        data = pickle.load(f)
        tele = int(input('Enter Student Roll no.: '))
        for i in data:
            if i[0] == tele:
                print(f'Name corresponds to {i[0]} is {i[1]}')
                k = False
        if k:
            print('Data not found!!')


while True:
    print('''
1. Add data.
2. To Search data.
3. Exit.
    ''')
    options = int(input('Enter a option(1-3):'))
    if options == 1:
        Add_data()
    elif options == 2:
        Show_data()
    else:
        break
```

```
1. Add data.
2. To Search data.
3. Exit.

Enter a option(1-3):1
Enter how many record to add: 3
Enter Student Roll no.: 1
Enter Student Name: ujjwal
Enter Student Roll no.: 2
Enter Student Name: garvit
Enter Student Roll no.: 3
Enter Student Name: mrk

1. Add data.
2. To Search data.
3. Exit.

Enter a option(1-3):2
Enter Student Roll no.: 2
Name corresponds to 2 is garvit

1. Add data.
2. To Search data.
3. Exit.

Enter a option(1-3):3
```

28.

```
# Create a binary file with roll number,
# name and marks. Input a roll number
# and update the marks.

f = open('s1', 'w+b')
n = int(input('Enter number of students: '))
for i in range(n):
    rollno = input('Enter rollno: ')
    name = input('Enter name of student: ')
    marks = input('Enter marks: ')
    bname = bytes(name, encoding='utf-8')
    brollno = bytes(rollno, encoding='utf-8')
    bmarks = bytes(marks, encoding='utf-8')
    f.write(brollno)
    f.write(bname)
    f.write(bmarks)
    f.write(b'\n')

f.seek(0)
data = f.read()        You, 2 days ago • code added
f.seek(0)
sk = input('Enter the roll no whose marks need updatation: ')
bsk = bytes(sk, encoding='utf-8')
l = len(bsk)
loc = data.find(bsk)
if loc < 0:
    print('Details not present')
else:
    f.seek(loc+l, 0)
    i = 0
    while f.read(1).isalpha():
        i = i+1
    f.seek(-1, 1)
    marksu = input('Enter updated marks: ')
    bmarksu = bytes(marksu, encoding='utf-8')
    f.write(bmarksu)
    print("Done")
```

```
Enter number of students: 2
Enter rollno: 1
Enter name of student: ujjwal
Enter marks: 200
Enter rollno: 2
Enter name of student: garvit
Enter marks: 190
Enter the roll no whose marks need updatation: 2
Enter updated marks: 200
Done
```

29.

```
# Remove all the lines that contain the character
# 'a' in a file and write it to another file.


with open("hp.txt", "w")as fo:
    fo.write("Harry Potter \n")
    fo.write("There is a difference in all harry potter books\n")
    fo.write('We can see it as harry grows\n')
    fo.write('the books were written by J.K rowling')

fo = open('hp.txt', 'r')
fi = open('writehp.txt', 'w')
l = fo.readlines()
for i in l:
    if 'a' in i:
        i = i.replace(i, '')
        fi.write(i)
    else:
        fi.write(i)
fi.close()
fo.close()
print('Done')
```

30.

```python
# Write a python program to search and display the
# record of the student from a binary file "Student.dat"
# containing students records (Rollno, Name and Marks).
# The user will enter roll number of the student to be searched.


import pickle


def set_data():
    rollno = int(input('Enter roll number: '))
    name = input('Enter name: ')
    test_marks = int(input('Enter test marks: '))
    print()
    student = {}
    student['rollno'] = rollno
    student['name'] = name
    student['test_marks'] = test_marks
    return student


def display_data(student):
    print('Roll number:', student['rollno'])
    print('Name:', student['name'])
    print('Test marks:', student['test_marks'])
    print()
      You, 2 days ago • code added

def write_record():
    outfile = open('student.dat', 'ab')
    pickle.dump(set_data(), outfile)
    outfile.close()


def read_records():
    infile = open('student.dat', 'rb')
    while True:
        try:
            student = pickle.load(infile)
            display_data(student)
        except EOFError:
            break
    infile.close()
```

```
def search_record():
    infile = open('student.dat', 'rb')
    rollno = int(input('Enter rollno to search: '))
    flag = False
    while True:
        try:
            student = pickle.load(infile)
            if student['rollno'] == rollno:
                display_data(student)
                flag = True
                break
        except EOFError:
            break

    if flag == False:
        print('Record not Found')
        print()
    infile.close()


while(True):
    print('''
        Menu
1. Add Record
2. Display Records
3. Search a Record
4. Exit
    ''')
    choice = input('Enter choice(1-4): ')
    print()
    if choice == '1':
        write_record()
    elif choice == '2':
        read_records()
    elif choice == '3':
        search_record()
    elif choice == '4':
        break
    else:
        print('Invalid input')
```

```
        Menu
1. Add Record
2. Display Records
3. Search a Record
4. Exit

Enter choice(1-4): 1

Enter roll number: 1
Enter name: ujjwal
Enter test marks: 200


        Menu
1. Add Record
2. Display Records
3. Search a Record
4. Exit

Enter choice(1-4): 1

Enter roll number: 2
Enter name: garvit
Enter test marks: 200

        Menu
1. Add Record
2. Display Records
3. Search a Record
4. Exit

Enter choice(1-4): 2

Roll number: 1
Name: ujjwal
Test marks: 200

Roll number: 2
Name: garvit
Test marks: 200


        Menu
1. Add Record
2. Display Records
3. Search a Record
4. Exit

Enter choice(1-4): 3

Enter rollno to search: 2
Roll number: 2
Name: garvit
Test marks: 200
```

31.

```
# WAP using function merge(file1,file2) to merge two
# files and obtain a third file called "NEW.DAT".
# The file1 and file2 are two files, which are pass
# as arguments. Also, display the content of third file.

import pickle


def merge(file1, file2, file3):
    with open(file1, 'rb') as f1, open(file2, 'rb') as f2:
        data_1 = pickle.load(f1)
        data_2 = pickle.load(f2)
    with open(file3, 'ab') as f:
        pickle.dump(data_1, f)
        pickle.dump('\n', f)
        pickle.dump(data_2, f)


file1 = input('Enter file1 name(with extention): ')
file2 = input('Enter file2 name(with extention): ')
file3 = input('Enter New file name(with extention): ')
merge(file1, file2, file3)
```

```
Enter file1 name(with extention): file1.dat
Enter file2 name(with extention): file2.dat
Enter New file name(with extention): file3.dat
```

32.

```
# Write a menu driven program where you are getting product
# related details in a file product.txt. menu options are
# 1. Add Details, 2. Search Details, 3. Show Details, 4. Exit

def add():
    with open('product.txt', "a") as f:
        a = int(input('Enter how many record to add: '))
        for _ in range(0, a):
            id = int(input('Enter product id: '))
            name = input('Enter product name: ')
            temp = (id, name)
            f.write(f'{temp} \n')
    print('product added successfully')
    input('Press ENTER to continue... ')

def see():
    with open("product.txt", "r") as f:
        d = f.readlines()
        d = [x.strip('\n') for x in d]
        for i in d:
            print(i)
    input('Press ENTER to continue... ')

def Search():
    with open("product.txt", "r")as f:
        d = f.readlines()
        d = [eval(x.strip('\n')) for x in d]
    name = input('Enter product name to Search: ')
    for i in range(0, len(d) - 1):
        if d[i][1] == name:
            print(f'productinfo is {d[i]}')

k = True
while k == True:
    print('''
1. Add Details
2. Search Details
3. Show Details
4. Exit
    ''')
    option = int(input('Enter your option(1/4): '))
    if option == 1:
        add()
    elif option == 2:
        Search()
    elif option == 3:
        see()
    elif option == 4:
        k = False
    else:
        print("Invalid Option!")
        continue
```

```
1. Add Details
2. Search Details
3. Show Details
4. Exit

Enter your option(1/4): 1
Enter how many record to add: 3
Enter product id: 1
Enter product name: samsung
Enter product id: 2
Enter product name: oneplus
Enter product id: 3
Enter product name: apple
product added successfully
Press ENTER to continue...

1. Add Details
2. Search Details
3. Show Details
4. Exit

Enter your option(1/4): 2
Enter product name to Search: oneplus
productinfo is (2, 'oneplus')

1. Add Details
2. Search Details
3. Show Details
4. Exit

Enter your option(1/4): 3
(1, 'samsung')
(2, 'oneplus')
(3, 'apple')
Press ENTER to continue...

1. Add Details
2. Search Details
3. Show Details
4. Exit

Enter your option(1/4): 4
```

33.

```python
# WAP to delete the file(s), which you want, should
# no longer exist in your computer. The file name
# should be entered at the time of program execution.
# If the entered file name not exist, display
# a proper message on screen

import os

file_name = input("Enter file name(with file extention): ")

if os.path.exists(file_name):
    os.remove(file_name)
else:
    print('File not found')
```

34.

```python
# A file emp.dat contains data attributes like : ecode, name and salary.
# Give function definitions to do the following
# a) Write the data of an employee.
# b) Read the employee data and display all the
# objects on the screen where salary is between 20000 and 30000.

import pickle

global global_flag, data_1
global_flag = False
data_1 = []


def Add_data():
    data_2 = []
    with open("emp.dat", 'ab+') as f:
        if global_flag == True:
            f.truncate(0)
        a = int(input('Enter how many record to add: '))
        for _ in range(0, a):
            ecode = int(input('Enter ecode: '))
            name = input("Enter Name: ")
            salary = int(input("Enter Salary: "))
            temp = [ecode, name, salary]
            data_1.append(temp)
        data_2.append(data_1)
        pickle.dump(data_2, f)
    f.close()


def Show_data():
    flag = True
    with open("emp.dat", 'rb') as f:
        data = pickle.load(f)
        for j in range(0, len(data)):
            for i in range(0, len(data[j])):
                if 20000 <= data[j][i][2] <= 30000:
                    print(f'Ecode = {data[j][i][0]}')
                    print(f'Name = {data[j][i][1]}')
                    print(f'Salary = {data[j][i][2]}')
                    flag = False
        if flag == True:
            print('No Entry Found!!!')
```

```python
while True:
    print('''
    1. Add data.
    2. To show data.
    3. Exit.
    ''')
    options = int(input('Enter a option(1-3):'))
    if options == 1:
        Add_data()
        global_flag = True
    elif options == 2:
        Show_data()
    else:
        break
```

```
    1. Add data.
    2. To show data.
    3. Exit.

Enter a option(1-3):1
Enter how many record to add: 2
Enter ecode: 1
Enter Name: ujjwal
Enter Salary: 25000
Enter ecode: 2
Enter Name: garvit
Enter Salary: 35000

    1. Add data.
    2. To show data.
    3. Exit.

Enter a option(1-3):2
Ecode = 1
Name = ujjwal
Salary = 25000

    1. Add data.
    2. To show data.
    3. Exit.

Enter a option(1-3):3
```

35.

```
# Write a menu driven program to add and manipulate data
# from customer.csv file. Give function to do the following:
# 1.Add Customer Details, 2.Search Customer Details
# 3.Remove Customer Details
# 4.Display all the Customer Details, 5.Exit


import csv


def add():
    with open('customer.csv', 'w', newline='') as f:
        writer = csv.writer(f)
        writer.writerow(["Cus_no", "C_name"])
        n = int(input('Enter how many Customer you want to insert: '))
        for _ in range(0, n):
            cus_no = int(input('Enter Customer ID: '))
            cus_name = input('Enter Customer Name: ')
            info = [cus_no, cus_name]
            writer.writerow(info)


def display():
    with open('customer.csv', 'r') as f:
        reader = csv.reader(f)
        for row in reader:
            print(row)


def search():
    data = []
    with open('customer.csv', 'r') as f:
        reader = csv.reader(f)
        for row in reader:
            data.append(row)
    id = input('Enter customer ID to search info: ')
    for i in data:
        if i[0] == id:
            print(f'Customer no. {i[0]} and Name is {i[1]}')
```

```python
def remove():
    data = []
    new_data = []
    with open('customer.csv', 'r') as f:
        reader = csv.reader(f)
        for row in reader:
            data.append(row)
    print(data)
    name = input('Enter customer name to delete info: ')
    with open('customer.csv', 'w', newline='') as f:
        f.truncate(0)
        for i in data:
            if i[1] ≠ name:
                new_data.append(i)
        writer = csv.writer(f)
        writer.writerows(new_data)


k = True
while k == True:
    print('''      You, 2 days ago • code added
1.Add Customer Details
2.Search Customer Details
3.Remove Customer Details
4.Display all the Customer Details
5.Exit
    ''')
    option = int(input('Enter your option(1/5): '))
    if option == 1:
        add()
    elif option == 2:
        search()
    elif option == 3:
        remove()
    elif option == 4:
        display()
    elif option == 5:
        k = False
    else:
        print("Invalid Option!")
        continue
```

```
1.Add Customer Details
2.Search Customer Details
3.Remove Customer Details
4.Display all the Customer Details
5.Exit

Enter your option(1/5): 1
Enter how many Customer you want to insert: 3
Enter Customer ID: 1
Enter Customer Name: ujjwal
Enter Customer ID: 2
Enter Customer Name: garvit
Enter Customer ID: 3
Enter Customer Name: mrk

1.Add Customer Details
2.Search Customer Details
3.Remove Customer Details
4.Display all the Customer Details
5.Exit

Enter your option(1/5): 2
Enter customer ID to search info: 3
Customer no. 3 and Name is mrk

1.Add Customer Details
2.Search Customer Details
3.Remove Customer Details
4.Display all the Customer Details
5.Exit

Enter your option(1/5): 3
[['Cus_no', 'C_name'], ['1', 'ujjwal'], ['2', 'garvit'], ['3', 'mrk']]
Enter customer name to delete info: garvit

1.Add Customer Details
2.Search Customer Details
3.Remove Customer Details
4.Display all the Customer Details
5.Exit

Enter your option(1/5): 4
['Cus_no', 'C_name']
['1', 'ujjwal']
['3', 'mrk']

1.Add Customer Details
2.Search Customer Details
3.Remove Customer Details
4.Display all the Customer Details
5.Exit

Enter your option(1/5): 5
```

36.

```
# Write a program to create a queue called Doctor to perform the
# basic operations on queue using list. The list contains two
# data fields: Docid and Docname. Write the following functions:
# InsertDoc() – To push the data values into the list  Docinfo
# DeleteDoc() – To remove the data value from the list  Docinfo
# ShowDoc(): – To display data value for all Docinfo.

# Queue
Doctor = []


def InsertDoc():
    n = int(input('Enter who many Doctor you want to insert: '))
    for _ in range(0, n):
        Docid = int(input('Enter Doctor ID: '))
        Docname = input('Enter Doctor: ')
        Docinfo = (Docid, Docname)
        Doctor.append(Docinfo)


def DeleteDoc():
    Doctor.pop(0)
    print('Done')


def ShowDoc():
    print(Doctor)


k = True
while k == True:
    print('''
1. Push Values
2. Delete Values
3. See Values
4. Exit
    ''')
    option = int(input('Enter your option(1/4): '))
    if option == 1:
        InsertDoc()
    elif option == 2:
        DeleteDoc()
    elif option == 3:
        ShowDoc()
    elif option == 4:
        k = False
    else:
        print("Invalid Option!")
        continue
```

```
1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 1
Enter who many Doctor you want to insert: 3
Enter Doctor ID: 1
Enter Doctor: ujjwal
Enter Doctor ID: 2
Enter Doctor: garvit
Enter Doctor ID: 3
Enter Doctor: bala

1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 2
Done

1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 3
[(2, 'garvit'), (3, 'bala')]

1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 4
```

37.

```
# Write two functions queins() to insert and quedel() to delete elements
# for customeer information, i.e. custno, cname using list

customer = []


def queins():
    n = int(input('Enter who many customer you want to insert: '))
    for _ in range(0, n):
        custno = int(input('Enter customer ID: '))
        cname = input('Enter customer Name: ')
        temp = [custno, cname]
        customer.append(temp)


def quedel():
    custno = int(input('Enter customer ID you want to delete: '))
    for i in range(0, len(customer)-1):
        if customer[i][0] == custno:
            del customer[i]


k = True
while k == True:
    print('''
1. Add Customer
2. Delete Customer
3. See Customer
4. Exit
    ''')
    option = int(input('Enter your option(1/4): '))
    if option == 1:
        queins()
    elif option == 2:
        quedel()
    elif option == 3:
        print(customer)
    elif option == 4:
        k = False
    else:
        print("Invalid Option!")
        continue
```

```
1. Add Customer
2. Delete Customer
3. See Customer
4. Exit

Enter your option(1/4): 1
Enter who many customer you want to insert: 3
Enter customer ID: 1
Enter customer Name: ujjwal
Enter customer ID: 2
Enter customer Name: bala
Enter customer ID: 3
Enter customer Name: sid

1. Add Customer
2. Delete Customer
3. See Customer
4. Exit

Enter your option(1/4): 2
Enter customer ID you want to delete: 2

1. Add Customer
2. Delete Customer
3. See Customer
4. Exit

Enter your option(1/4): 3
[[1, 'ujjwal'], [3, 'sid']]

1. Add Customer
2. Delete Customer
3. See Customer
4. Exit

Enter your option(1/4): 4
```

38.

```
# Write a program to create a stack called Product to perform
# the basic operations on stack using list. The list contains
# two data fields: ProductId and ProductName. Write the following functions:
# InsertProd() – To push the data values into the list  Docinfo
# DeleteProd() – To remove the data value from the list  Docinfo
# ShowProd(): – To display data value for all Docinfo.

# stack
product = []


def InsertProd():
    n = int(input('Enter who many products you want to insert: '))
    for _ in range(0, n):
        pro_id = int(input('Enter product ID: '))
        pro_name = input('Enter product: ')
        Docinfo = (pro_id, pro_name)
        product.append(Docinfo)


def DeleteProd():
    product.pop()
    print('Done')


def ShowProd():
    print(product)


k = True
while k == True:
    print('''
1. Push Docinfo
2. Delete Docinfo
3. See Product
4. Exit
    ''')
    option = int(input('Enter your option(1/4): '))
    if option == 1:
        InsertProd()
    elif option == 2:
        DeleteProd()
    elif option == 3:
        ShowProd()
    elif option == 4:
        k = False
    else:
        print("Invalid Option!")
        continue
```

```
1. Push Docinfo
2. Delete Docinfo
3. See Product
4. Exit

Enter your option(1/4): 1
Enter who many products you want to insert: 3
Enter product ID: 1
Enter product: mobile
Enter product ID: 2
Enter product: tab
Enter product ID: 3
Enter product: computer

1. Push Docinfo
2. Delete Docinfo
3. See Product
4. Exit

Enter your option(1/4): 2
Done

1. Push Docinfo
2. Delete Docinfo
3. See Product
4. Exit

Enter your option(1/4): 3
[(1, 'mobile'), (2, 'tab')]

1. Push Docinfo
2. Delete Docinfo
3. See Product
4. Exit

Enter your option(1/4): 4
```

39.

```python
# Write a menu-based program to perform the operation on queue in python.

queue = []


def Insert():
    n = int(input('Enter how many no. you want to insert: '))
    for _ in range(0, n):
        value = int(input('Enter no.: '))
        queue.append(value)


def Delete():
    queue.pop(0)
    print('Done')


def Show():
    print(queue)


k = True
while k == True:
    print('''
1. Push Values
2. Delete Values
3. See Values
4. Exit
    ''')
    option = int(input('Enter your option(1/4): '))
    if option == 1:
        Insert()
    elif option == 2:
        Delete()
    elif option == 3:
        Show()
    elif option == 4:
        k = False
    else:
        print("Invalid Option!")
        continue
```

```
1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 1
Enter how many no. you want to insert: 3
Enter no.: 12
Enter no.: 32
Enter no.: 14

1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 2
Done

1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 3
[32, 14]

1. Push Values
2. Delete Values
3. See Values
4. Exit

Enter your option(1/4): 4
```

40.

a. SHOW DATABASES;
b. CREATE DATABASE office.
c. SHOW TABLES;
d. CREATE TABLE emp(
   Empno int,
   Empname varchar(30),
   Desig varchar(30),
   Hiredate date,
   Salary int,
   Deptno int)
e. ALTER TABLE emp ADD Mobile int, email varchar(50);
f. ALTER TABLE emp DROP COLUMN Mobile;
g. DESC emp;
h. DROP office.emp;

41.

a.
   insert into employee(emp_id, emmp_name, age, phone_num, dept_id)
   values(1,"ujjwal",18,1305240266,1),
   values(2,"garvit",13,1125723767,1),
   values(3,"mrk",24,0708474991,1),
   values(4,"harsh",15,2377720050,1),
   values(5,"muskan",13,0370602270,1),
   values(6,"amrit",15,6051842944,1),
   values(7,"sukarn",14,3411679960,1),
   values(8,"shailaja",18,8698480510,1),
   values(9,"akshay",17,0467788858,1),
   values(10,"kashish",16,0467788858,1);
b. SELECT *FROM emp_table;
c. UPDATE emp_table SET salary = salary + 50;
d. DELETE FROM emp_table WHERE department_no = 50;

42.

a. SELECT DISTINCT department FROM Employee;
b. SELECT Emp_name, Salary FROM Employee WHERE Salary BETWEEN 35000 AND 40000;
c. SELECT Emp_name FROM Employee WHERE city = 'Guwahati' or city = 'Surat' or city = 'Jaipur';
d. SELECT Emp_name FROM Employee WHERE Emp_name LIKE 'M%';
e. SELECT Emp_name FROM Employee WHERE department IS NULL;
f. SELECT * FROM employees ORDER BY emp_id ASC;
g. SELECT department, AVG(salary) FROM employees GROUP BY department;
h. NOT DONE

43.

a. SELECT avg(salary) FROM employees;

b. SELECT MIN(salary) FROM employees;

c. SELECT MAX(salary) FROM employees;

d. SELECT SUM(salary) FROM employees WHERE city = 'Guwahati';

44.

CARTESIAN JOIN:

Syntax:

SELECT table1.column1 , table1.column2, table2.column1...

FROM table1

CROSS JOIN table2;

table1: First table.

table2: Second table

EQUI JOIN:

Syntax:

SELECT column_list

FROM table1, table2....

WHERE table1.column_name =

table2.column_name;

45.

```
# Write a program to connect Python with MySQL using database
# connectivity and perform the following operations on data
# in database: Fetch, Update and delete the data

import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="Ujjw@l.16",
    charset='utf8')
cur = mydb.cursor()
cur.execute('create database if not exists parctical')
cur.execute('use parctical')
mydb.commit()

table = '''
create table if not exists file_table(
name varchar(50),
address varchar(50),
phone_no varchar(30))
'''
cur.execute(table)
mydb.commit()


def add_Details():
    print('Enter the details :')
    s = 'insert into file_table (name,address,phone_no)' \
        'values(%s,%s,%s) '
    name = input('Enter your name: ')
    address = input('Enter your address: ')
    phone_no = input('Enter your phone number: ')
    value = (name, address, phone_no)
    cur.execute(s, value)
    mydb.commit()
    print("Successfully updated!!!!!!!!")


def delete():
    item_id = input('Enter Name that you want to delete: ')
    s = 'DELETE FROM file_table WHERE name =%s'
    value = (item_id,)
    cur.execute(s, value)
    mydb.commit()
    print('Successfully deleted!!!!!!')
    input('Press ENTER to continue.....')
```

```
def update():
    cur.execute('select * from file_table')
    result = cur.fetchall()
    for rec in result:
        print(rec)
    s = "update file_table set address = %s, phone_no = %s where name = %s"
    name = input('Enter name for which you want to update: ')
    address = input('Enter address: ')
    phone_no = input('Enter phone number: ')
    value = (address, phone_no, name)
    cur.execute(s, value)
    mydb.commit()
    print("Successfully updated!!!!!!!!")


def see_details():
    s = 'select * from file_table'
    cur.execute(s)
    result = cur.fetchall()
    for rec in result:
        print(rec)
    input('Press ENTER to continue.....')


k = True
while k == True:
    print('''
1.Add Details
2.Update Details
3.Delete Details
4.See Details
5.Exit''')
    option = int(input('Enter your Options(1/5): '))
    if 0 <= option <= 4:
        if option == 1:
            add_Details()
        elif option == 2:
            update()
        elif option == 3:
            delete()
        elif option == 4:
            see_details()
    elif option == 5:
        k = False
    else:
        print('Invalid Option!')
        continue
```

S

```
1.Add Details
2.Update Details
3.Delete Details
4.See Details
5.Exit
Enter your Options(1/5): 1
Enter the details :
Enter your name: ujjwal
Enter your address: dps mk
Enter your phone number: 1234
Successfully updated!!!!!!!!

1.Add Details
2.Update Details
3.Delete Details
4.See Details
5.Exit
Enter your Options(1/5): 1
Enter the details :
Enter your name: bala
Enter your address: sector 44
Enter your phone number: 123456
Successfully updated!!!!!!!!

1.Add Details
2.Update Details
3.Delete Details
4.See Details
5.Exit
Enter your Options(1/5): 2
('ujjwal', 'dps mk', '1234')
('bala', 'sector 44', '123456')
Enter name for which you want to update: ujjwal
Enter address: dps mk
Enter phone number: 123456789
Successfully updated!!!!!!!!

1.Add Details
2.Update Details
3.Delete Details
4.See Details
5.Exit
Enter your Options(1/5): 3
Enter Name that you want to delete: bala
Successfully deleted!!!!!!
Press ENTER to continue.....

1.Add Details
2.Update Details
3.Delete Details
4.See Details
5.Exit
Enter your Options(1/5): 4
('ujjwal', 'dps mk', '123456789')
Press ENTER to continue.....
```